Design and initial in-water testing of advanced non-linear control algorithms onto an Unmanned Underwater Vehicle (UUV)

Vladimir Djapic
Unmanned Maritime Vehicle Lab
SPAWAR Sys Cen - San Diego
San Diego, CA 92152, USA
Dept of Electrical Eng
UC Riverside
Riverside, CA 92521, USA
Email: djapic@spawar.navy.mil

J.A. Farrell
Dept of Electrical Eng
UC Riverside
Riverside, CA 92521, USA
Email: farrell@ee.ucr.edu

Paul Miller
Unmanned Maritime Vehicle Lab
SPAWAR Sys Cen - San Diego
San Diego, CA 92152, USA
Email: pamiller@spawar.navy.mil

Rich Arrieta
Unmanned Maritime Vehicle Lab
SPAWAR Sys Cen - San Diego
San Diego, CA 92152, USA
Email: rich.arrieta@navy.mil

***Abstract*— UUVs equipped with high-level control software have a variety of potential applications for Anti-Terrorism/Force Protection (ATFP) objectives. Desirable vehicle control capabilities include the ability to drive at very low, controllable speeds, the ability to maintain a set distance and attitude (pitch and roll) relative to some surface for optimal sensor (both sonar and video) effectiveness, and the ability for the operator to intervene to change the mission activities. Moreover, a vehicle capable of rotating in place or having a fraction of a meter turning radius is needed to conduct the mission. The present state-of-art vehicles are not maneuverable enough to successfully accomplish most of these tasks. The novel controllers are expected to be nonlinear due to the fact that the vehicle is translating at nonzero attitude or translating in a direction different from that of the surface. Non-linear controller that compensates for non-linear forces (such as drag, buoyancy/gravity) was designed, installed onto the UUV test-bed, and in-water tested. The structure of a controller is hierarchical such that an "inner loop" non-linear controller (outputs the appropriate thrust values) is the same for all mission scenarios while an appropriate "outer-loop" non-linear controller is used based on what mission or part of the mission is desired.**

I. Introduction

An increasing variety of sensors are becoming available for use onboard autonomous vehicles. Given these enhanced sensing capabilities, scientific and military personnel are interested in exploiting autonomous vehicles for increasingly complex missions. Most of these missions require the vehicle to function in complex, cluttered environments and react to changing environmental parameters. This paper presents the control design, simulation results, and initial in-water results of several complex mission scenarios. One of the primary mission, for which the novel controller was used, was a mission where the vehicle was using the data from onboard sonar Acoustic Doppler Current Profiler (ADCP) sensor in order to accurately follow a curved surface below it. The sonar outputs four beams which measure the distance from the vehicle to the bottom, as well as, the relative vehicle attitude to the bottom. The vehicle was tested following the bottom of the 300 ft by 200 ft by 38 ft deep Transducer Evaluation Center (TRANSDEC) pool at SPAWAR Systems Center San Diego (SSC-SD). The pool is bowl shaped so its depth is increasing toward the center of it. The vehicle's task was to intelligently maintain the desired stand-off distance from the curved bottom by adjusting its pitch and roll based on the information coming from the sonar sensor. Another task for the vehicle was to track a line along the desired path while maintaining the desired distance (altitude) from the bottom. The vehicle used the novel translational, attitude, and altitude controller based on the nonlinear control technique called Backstepping. The novel controller implements the ideas of Backstepping and, in addition, introduces a new algorithm called Command Filtered (CF) Backstepping.

In this paper, we also present the novel vehicle simulator which simulates both the navigation and control systems of the vehicle. The simulator is in the form of open-architecture which allows other, additional controllers to be easily added. This simulator can simulate any mission from start to end allowing the operator the chance to check, review, and possibly change the mission plan prior to loading the mission onto the UUV. In essence, it gives the operator the assurance that the mission will result in a success. The actual in-water results very closely resembled the simulated mission for all control modes tested. This paper stresses the comparison between the simulated and actual in-water mission performed by the UUV.

Several places in this paper refer to filtering of a signal x_c^o to produce a signal x_c and its derivative $\dot{x}_c$. This is referred to as command filtering. The motivation of command filtering is to determine the signals $x_c(t)$ and $\dot{x}_c(t)$ as needed for the next iteration of the backstepping procedure [7], without having to analytically differentiate x_c^o, because the analytic differentiation becomes overly cumbersome for systems of

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2007		**00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Design and Initial In-Water Testing of Advanced Non-Linear Control Algorithms Onto an Unmanned Underwater Vehicle (UUV)	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Unmanned Maritime Vehicle Lab,SPAWAR Sys Cen - San Diego,San Diego,CA,92152**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
MTS/IEEE 2007, Vancouver, BC, Canada, Sept. 29 - Oct 4, 2007.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**6**	

high dimension. The effects of command filtering on the backstepping stability analysis are analyzed in [4], [5], [6] and will not be repeated herein. The summary of that analysis is that for a properly designed command filter (unity DC gain to the first output and the second output being the derivative of the first output) the system will be stable and the tracking error will be $\mathcal{O}\left(\frac{1}{\omega_B}\right)$ where ω_B is the bandwidth of the command filter. Therefore, the tracking error can be made arbitrarily small by increasing the parameter ω_B.

The paper is organized as follows. Section II describes the developed vehicle simulator and overall vehicle navigation and control structure. Section III defines a thruster powered UUV kinematics and dynamics equations. Section IV outlines the control law signals that need to be implemented for trajectory tracking and attitude control. Sections V, VI, and VII present a detailed derivation of trajectory tracking control laws to deal with vehicle kinematics and dynamics. The performance of the control system proposed is illustrated in simulation in Section VIII and in real in-water tests in Section IX. Finally, Section X contains the conclusions and describes some problems that warrant further research.

II. Vehicle Simulator and Overall Architecture

During this research effort we leveraged the existing vehicle sensors and improved their use by introducing novel control and navigation techniques. Vehicle dynamics, its sensors, and the environmental factors, such as currents, are modeled in a comprehensive vehicle simulation described in Figure 1. Our software simulates sensor noise and performance characteristics, range measurements, acoustic angle of incidence and line-of-sight requirements, and random drop outs. Simulation environment can import 3D models of arbitrary solid objects, such as ship hulls, sea floor terrain maps, quay walls, pier pilings, etc. The same vehicle and control software that is executed in simulation is executed onboard of the UUV. This approach significantly reduces costly in-water testing requirements as well as provides mission plan verification. This approach accelerates vehicle navigation, control, and mission development since we can experiment with operational challenges without asset risk. Moreover, the simulator can be used as a great operator training tool through basic operator tele-operation training. UUVs equipped with this type of software can greatly enhance current underwater security capabilities, relieving divers of time-consuming, dangerous tasks, therefore, reducing manpower and mission timeline requirements.

The overall vehicle navigation and control architecture is described in in Figure 2. Vehicle sensors, such as Inertial Measurement Unit (IMU), Doppler Velocity Log (DVL), Long-Baseline (LBL), compass, pressure, and altimeters, provide noisy measurements of the vehicle's movements. Navigation estimates the vehicle's true state (position, attitude, velocity, and angular rates). A Kalman filtering algorithm is implemented for vehicle navigation. The navigation software is such that, the fast rate sensor IMU is aided with slower rate DVL and LBL sensors in order to accurately navigate in the harsh environment, for example, under the hull of a ship in the harbor. Adaptive Mission Planner (AMP) manipulates the mission plan in order to optimally obtain a mission goal. Mission Spooler executes a series of behavioral commands dictated by the AMP. The hydro-forces acting on the vehicle are modeled and their effect is accounted for with the advanced nonlinear control technique. The expected model uncertainties are compensated for using a novel approach to backstepping nonlinear control technique. Great tracking of the desired trajectory is achieved. The structure of a controller is hierarchical such that an "inner loop" non-linear controller (outputs the appropriate thrust values) is the same for all mission scenarios while an appropriate "outer loop" nonlinear controller is used based on what mission or part of the mission is desired. Inner loop controller computes the desired thrust values to achieve the commanded velocities and angular rates and sends the commands to thrusters which create vehicle movement. Outer loop controller computes the desired velocities and angular rates to achieve a behavior.

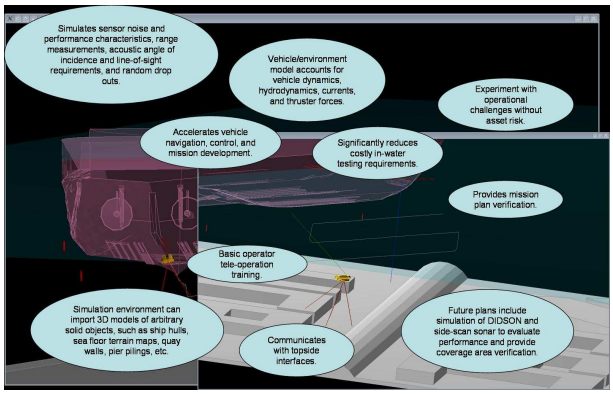

Fig. 1. Comprehensive Vehicle Simulation

Using the described vehicle architecture structure and the approach of testing the entire vehicle mission in simulation prior to performing the actual in-water test we achieved great results at the recent Autonomous Underwater Vehicle Festival (AUVFest), June 6-15 2007, held in Panama City. The vehicle employed for simulation and in-water testing is underactuated since the lateral speed v is not directly affected by the thrusters (no side thrusters), which makes certain desired maneuvers more difficult to achieve. During AUV Fest, we accomplished 12 hours of in-water demonstration time. All hull search vehicle behaviors were demonstrated with the following unique capabilities: autonomous mission execution with intervention capability, hull search conducted using side look sonar, sensors parameters optimized by operator during mission, real-time topside display of Sound Metrics DIDSON High Definition Imaging Sonar and Marine Sonics 1800kHz Side Looking Sonar, vehicle position and status information embedded in DIDSON sensor data and Joint Architecture for Unmanned Systems (JAUS) communication protocol implemented on UUV.

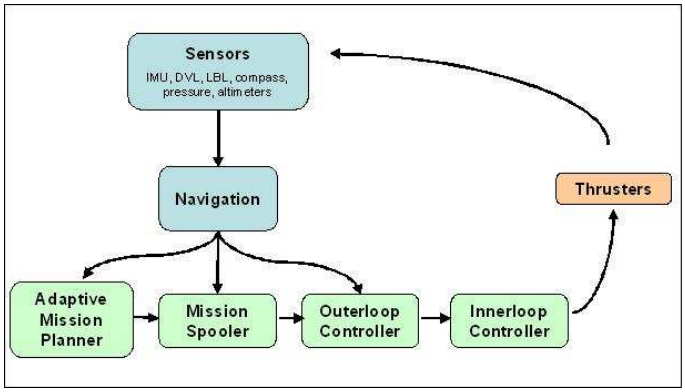

Fig. 2. Navigation and Control Architecture

III. THRUSTER POWERED AUV DYNAMICS

Let the vehicle dynamics be described as [2]

$$\begin{aligned} \dot{p} &= R_b^t v_b && (1)\\ \dot{\Theta} &= \Omega\omega && (2)\\ \dot{v}_b &= M^{-1}(u_1 - F_{nlin}) && (3)\\ \dot{\omega} &= J^{-1}(u_2 - M_{nlin}) && (4) \end{aligned}$$

where $p = [x, y, d]$ is the earth relative position, R_b^t is the rotation from body to earth frame, $v_b = [u, v, w]^\top$ is the velocity in body frame, $\Theta = [\phi, \theta, \psi]$ is the attitude, Ω is a nonlinear (nonsingular except at $\theta = \frac{\pi}{2}$) matrix function of Θ, ω is the inertial rotation rate vector represented in body frame, F_{nlin} represents the body-frame nonlinear forces, M_{nlin} represents the body-frame nonlinear moments, u_1 is the vector of control forces, u_2 is the vector of control moments. The control forces and moments are generated by a set of five thrusters mounted to achieve full angular rate control (i.e., ω), surge control (i.e., u), and heave control (i.e., w). The vector $T = [T_1, \ldots, T_5]^\top$ of five thrusts is related to the the control forces and moments by a known thrust distribution matrix such that $u_1 = L_f T$ and $u_2 = L_m T$ where $L_f \in \Re^{2\times 5}$ and $L_m \in \Re^{3\times 5}$. The AUV is underactauted since the lateral speed v is not directly affected by the thrusters.

The rotation matrix, R_b^t, is defined as

$$R_b^t = \begin{bmatrix} c\theta c\psi & c\psi s\theta s\phi - c\phi s\psi & c\phi c\psi s\theta + s\phi s\psi \\ c\theta s\psi & c\phi c\psi + s\theta s\phi s\psi & -c\psi s\phi + c\phi s\theta s\psi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix},$$

and the angular rate transformation matrix, Ω, as

$$\Omega = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix},$$

where the symbols c_z, s_z, and t_z represent cos(z), sin(z), and tan(z).

IV. CONTROL SIGNALS IMPLEMENTATION

This section summarizes the control law. The stability analysis is rigorously analyzed in [3] as well as in the journal paper that the authors are working on [1]. Due to lack of space, the arguments are not repeated herein. The following equations represent the control signals

$$\begin{aligned} u_c^o &= \gamma \frac{\| \ [-F_x - K_{xy}\tilde{x} + \dot{x}_c, -F_y - K_{xy}\tilde{y} + \dot{y}_c] \ \|}{c\theta}\\ \psi_c^o &= atan2\left[\gamma\left(-F_y - K_{xy}\tilde{y} + \dot{y}_c\right), \gamma\left(-F_x - K_{xy}\tilde{x} + \dot{x}_c\right)\right]\\ w_c^o &= \frac{u\sin(\theta) - \cos(\theta)\sin(\phi)v - K_d\tilde{d} + \dot{d}_c}{\cos(\theta)\cos(\phi)}\\ \omega_c^o &= \Omega^{-1}\left(-K_\Theta\tilde{\Theta} + \dot{\Theta}_c - \Theta_{bs}\right)\\ u_1 &= M(F_{nlin} - K_v\tilde{v}_b + \dot{v}_{bc} - v_{bbs})\\ u_2 &= J(M_{nlin} - K_\omega\tilde{\omega} + \dot{\omega}_c - \omega_{bs}), \end{aligned}$$

where $\gamma = \pm 1$ and ψ_{bs}, u_{bs}, w_{bs},and ω_{bs} are defined in eqns. 9, 8, 13, and 16, respectively. Because in this article, the θ and ϕ commands are externally generated, while ψ is used as a control variable, only yaw backstepping term, ψ_{bs}, must be defined for implementation of ω_c^o signal, which is done in Section V. The terms θ_{bs} and ϕ_{bs} are identically zero. The term v_{bbs} is the vector notation of u_{bs} and w_{bs} terms while ω_{bs} is the vector notation of p_{bs}, q_{bs}, r_{bs} terms. For $\gamma = 1$ the vehicle drives forward while for $\gamma = -1$ the vehicle drives backward. Also, because this is trajectory following, it is assumed that the speed $\|(\dot{x}_c, \dot{y}_c)\|$ is non-zero and we have selected a solution for which the AUV forward velocity is always positive.

V. TRAJECTORY FOLLOWING

The inputs to this control loop are $x_c(t)$, $y_c(t)$, and the derivatives, $\dot{x}_c$, $\dot{y}_c$. We assume that

$$\left\| \begin{array}{c} \dot{x}_c \\ \dot{y}_c \end{array} \right\| \geq \epsilon > 0$$

This section is concerned with the control of $[x, y]$ by specification of desired values for $[u, \psi]$.

A. Kinematic Analysis

Since v is not controllable and w is used to control depth we will control x and y by calculating appropriate u_d and ψ_d signals. For clarity, we rewrite x and y dynamics as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} c\theta + \begin{bmatrix} F_x \\ F_y \end{bmatrix}$$

where

$$F_x = [c\psi s\theta s\phi - c\phi s\psi]v + [c\phi c\psi s\theta + s\phi s\psi]w,$$

$$F_y = [c\phi c\psi + s\theta s\phi s\psi]v + [-c\psi s\phi + c\phi s\theta s\psi]w,$$

and

$$\left.\begin{array}{l} u_x = u\ c\psi \\ u_y = u\ s\psi. \end{array}\right\} \quad (5)$$

The dynamic equation for x and y can be manipulated as follows

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \end{bmatrix} + \begin{bmatrix} u_{x_c}^o \\ u_{y_c}^o \end{bmatrix} c\theta + \begin{bmatrix} \tilde{u}_x \\ \tilde{u}_y \end{bmatrix} c\theta$$

where $\tilde{u}_x = u_x - u_{x_c}$ and $\tilde{u}_y = u_y - u_{y_c}$. Again, there is another term, $[u_{x_c} - u_{x_c}^o, u_{y_c} - u_{y_c}^o]^\top$, that should be accounted for in the analysis, where $u_{x_c} = u_c c\psi_c$, $u_{x_c}^o = u_{c_o} c\psi_c^o$, etc.

This term can be made arbitrarily small by increasing the bandwidth of the command filter that is used to compute u_c and ψ_c (and their derivatives) from u_c^o and ψ_c^o. The effect of this term is rigorously analyzed in [3]. Due to lack of space, the arguments are not repeated herein. We select signals $[u_{x_c}^o, u_{y_c}^o]^\top$ as

$$\begin{bmatrix} u_{x_c}^o \\ u_{y_c}^o \end{bmatrix} = \begin{bmatrix} \frac{1}{\cos(\theta)}(-F_x - K_{xy}\tilde{x} + \dot{x}_c) \\ \frac{1}{\cos(\theta)}(-F_y - K_{xy}\tilde{y} + \dot{y}_c) \end{bmatrix} \tag{6}$$

where K_{xy} is time varying and positive. The selection of the control signal above yields the x and y position error dynamic equations:

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \end{bmatrix} = \begin{bmatrix} -K_{xy}\tilde{x} \\ -K_{xy}\tilde{y} \end{bmatrix} + \begin{bmatrix} \tilde{u}_x \\ \tilde{u}_y \end{bmatrix} \mathrm{c}\theta \tag{7}$$

Two questions remain: how should we manipulate the $\tilde{u}_x$ and $\tilde{u}_y$ terms to allow a rigorous stability analysis that accounts for them; and, how should u_c^o and ψ_c^o be specified to achieve $[u_{x_c}^o, u_{y_c}^o]$? These issues are addressed in the following subsections.

Simplification of $\tilde{u}_x$ and $\tilde{u}_y$. These terms can be manipulated by two very similar approaches (derived in [1]). In either case the $\begin{bmatrix} \tilde{u}_x \\ \tilde{u}_y \end{bmatrix}$ term can be expressed in the following form

$$\begin{bmatrix} \tilde{u}_x \\ \tilde{u}_y \end{bmatrix} = A\tilde{u} + Bg(\tilde{\psi})\tilde{\psi}.$$

Thus, the position error dynamics can be expressed as

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \end{bmatrix} = \begin{bmatrix} -K_{xy}\tilde{x} \\ -K_{xy}\tilde{y} \end{bmatrix} + (A\tilde{u} + Bg(\tilde{\psi})\tilde{\psi})\cos(\theta)$$

which is a form suitable for stability analysis. The first term will be accommodated by the u backstepping control. The second term will be accommodated by the ψ backstepping control. Selecting Approach 1 in [1], we define u_{bs} and ψ_{bs} backstepping terms as

$$u_{bs} = \mathrm{c}\theta(\mathrm{c}\psi_c\tilde{x} + \mathrm{s}\psi_c\tilde{y}) \tag{8}$$

and

$$\begin{aligned} \psi_{bs} \quad = \quad & \mathrm{c}\theta \left\{ \left[u\ \mathrm{c}\psi_c \frac{\cos(\tilde{\psi}) - 1}{\tilde{\psi}} - u\ \mathrm{s}\psi_c \frac{\sin(\tilde{\psi})}{\tilde{\psi}} \right] \tilde{x} \right. \\ & \left. + \left[u\ \mathrm{s}\psi_c \frac{\cos(\tilde{\psi}) - 1}{\tilde{\psi}} + u\ \mathrm{c}\psi_c \frac{\sin(\tilde{\psi})}{\tilde{\psi}} \right] \tilde{y} \right\}. \end{aligned} \tag{9}$$

B. Selection of Control Gain

This subsection discusses technical details related to the selection of K_{xy} in eqn. (6).

A solution to eqn. (6) for u_c and ψ_c is

$$u_c^o = \frac{\| \ [-F_x - K_{xy}\tilde{x} + \dot{x}_c, -F_y - K_{xy}\tilde{y} + \dot{y}_c] \ \|}{\mathrm{c}\theta}.$$

and

$$\psi_c^o = atan2\left(-F_y - K_{xy}\tilde{y} + \dot{y}_c, -F_x - K_{xy}\tilde{x} + \dot{x}_c\right)$$

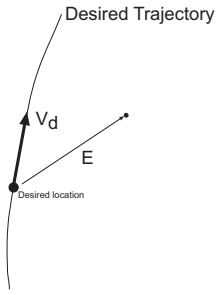

Fig. 3. Trajectory depiction

where we constrain the commanded value of θ such that $\theta \neq \frac{\pi}{2}$. Also, because this is trajectory following, it is assumed that the speed $\|(\dot{x}_c, \dot{y}_c)\|$ is non-zero and we have selected a solution for which the AUV forward velocity is always positive.

Eqn. (6) has the form,

$$\begin{bmatrix} u_c^o\ \mathrm{c}\psi_c^o \\ u_c^o\ \mathrm{s}\psi_c^o \end{bmatrix} = v_d - K_{xy}E$$

where

$$v_d = \begin{bmatrix} \frac{1}{\cos(\theta)}(-F_x + \dot{x}_c) \\ \frac{1}{\cos(\theta)}(-F_y + \dot{y}_c) \end{bmatrix} \text{ and } E = \begin{bmatrix} \frac{1}{\cos(\theta)}(\tilde{x}) \\ \frac{1}{\cos(\theta)}(\tilde{y}) \end{bmatrix}$$

The quantity v_d is the velocity vector that would cause the vehicle to follow the trajectory given that the vehicle was currently on the trajectory. The quantity $K_{xy}E$ is the feedback term that would cause the vehicle to converge toward the trajectory.

For the stability analysis to follow, the value of K_{xy} must be positive; however, its magnitude must be selected with care. Consider the situation depicted in Figure 3 where the inner product of v_d and E is positive (i.e., the AUV is ahead of the current desired trajectory position). Since the speed cannot be negative, depending on the value of K_{xy}, the commanded yaw angle could result in the vehicle circling to get to the desired location. In particular, when $K_{xy}\|E\| > \|v_d\|$, the vehicle may be commanded in a direction opposite to the desired direction of the trajectory.

To prevent this we must ensure that the angle between v_d and $v_d - K_{xy}E$ is less than $90\,\mathrm{deg}$:

$$\begin{aligned} \langle v_d, (v_d - K_{xy}E)\rangle \quad &\geq \quad 0 \qquad (10) \\ \|v_d\|^2 \quad &\geq \quad K_{xy}\langle v_d, E\rangle. \qquad (11) \end{aligned}$$

There are three possible cases:

1) $\langle v_d, E\rangle > 0$: This is the problematic case that could result in the vehicle pointing opposite to the desired velocity if K_{xy} is too big. The value of K_{xy} should be selected such that
$$K_{xy} \leq \frac{\|v_d\|^2}{\langle v_d, E\rangle}.$$
2) $\langle v_d, E\rangle = 0$: In this case, the value of K_{xy} does not matter.

3) $\langle v_d, E\rangle < 0$: In this case, any positive value of K_{xy} satisfies eqn. (11).

Therefore, the designer specifies positive constants $\bar{k}$ and $0 < \alpha < 1$. At each time instant,

$$K_{xy}(t) = \begin{cases} \bar{k} & \text{if} \quad \langle v_d, E\rangle \leq 0 \\ \min\left(\frac{\alpha\|v_d\|^2}{\langle v_d, E\rangle}, \bar{k}\right) & \text{if} \quad \langle v_d, E\rangle > 0. \end{cases}$$

In situations such as that in Figure 3, this approach results in the vehicle driving towards the trajectory with the tangential component small enough that the trajectory point will ultimately catch up to the AUV. In the case where the vehicle is on the trajectory directly in front of the desired location, this choice approach causes the vehicle to drive slower than the desired point is moving, in effect waiting for the desired location to catch up.

C. Mode: D Translation

The objective of this component of the outer loop is to select $w_c^o(t)$ to force $d(t)$ to converge to $d_c(t)$, where $d_c(t)$ and $\dot{d}_c(t)$ are known command signals.

Using the last row of the R matrix, the dynamics of d are

$$\dot{d} = -u\sin(\theta) + \cos(\theta)\sin(\phi)v + \cos(\theta)\cos(\phi)w.$$

Since desired values for u, ϕ, and θ are already specified and v is not controllable, assuming that $\theta \neq 90^\circ$ and $\phi \neq 90^\circ$, we will select the commanded value of

$$w_c^o = \frac{u\sin(\theta) - \cos(\theta)\sin(\phi)v - K_d\tilde{d} + \dot{d}_c}{\cos(\theta)\cos(\phi)}$$

for w to control d. This yields the closed loop depth error dynamic equation

$$\dot{\tilde{d}} = -K_d\tilde{d} + \mathrm{c}\theta\mathrm{c}\phi\tilde{w} + \mathrm{c}\theta\mathrm{c}\phi(w_c - w_c^o) \tag{12}$$

where the last term is dropped in the subsequent analysis due to space limitations, but can be analyzed rigorously by methods similar to those in [3].

Following the approach described in [1] we can define the backstepping term

$$w_{bs} = \mathrm{c}\theta\mathrm{c}\phi\tilde{d} \tag{13}$$

that will be incorporated into the w inner loop control signal to cancel the sign indefinite portion in the stability analysis.

VI. MIDDLE LOOP - ATTITUDE CONTROL

This control loop will be used by each of the outer loops and therefore it is described separately. The inputs to this outer loop are roll, pitch, and yaw commands, $\Theta_c = [\phi_c(t), \theta_c(t), \psi_c(t)]$ and the derivatives of these signals, which are produced by command filtering. Additional inputs are the roll, pitch, and yaw backstepping terms, Θ_{bs}.

For attitude control, based on eqn. (2), we define the signal

$$\omega_c^o = \Omega^{-1}\left(-K_\Theta\tilde{\Theta} + \dot{\Theta}_c - \Theta_{bs}\right)$$

where K_Θ is a positive definite matrix and $\tilde{\Theta}(t) = \Theta(t) - \Theta_c(t)$. Using this definition, the closed-loop tracking error corresponding to eqn. (2) is

$$\begin{aligned} \dot{\Theta} &= \Omega\omega_c^o + \Omega(\omega - \omega_c) + \Omega(\omega_c - \omega_c^o) \qquad (14) \\ &= -K_\Theta\tilde{\Theta} + \dot{\Theta}_c + \Omega\tilde{\omega} - \Theta_{bs} \\ \dot{\tilde{\Theta}} &= -K_\Theta\tilde{\Theta} + \Omega\tilde{\omega} - \Theta_{bs} \qquad (15) \end{aligned}$$

where the term $\Omega(\omega_c - \omega_c^o)$ is dropped after eqn. (14). This term can be made arbitrarily small by increasing the bandwidth of the command filter that is used to compute ω_c and $\dot{\omega}_c$ from ω_c^o. The effect of this term is rigorously analyzed in [3]. Due to lack of space, the arguments are not repeated herein. To compensate for the sign indefinite $\tilde{\omega}$ term in the stability analysis the term ω_{bs} in Section VII is defined as

$$\omega_{bs} = \Omega^\top\tilde{\Theta}. \tag{16}$$

VII. INNER LOOP

The inputs to the inner loop are u_c, $\dot{u}_c$, w_c, $\dot{w}_c$, ω_c, $\dot{\omega}_c$, u_{bs}, w_{bs},and ω_{bs}. Each of these input signals is defined by one of the middle or outer loops as will be described in the sequel. Each of the signals u_c, w_c, and ω_c are commands to the inner loop. The signals $\dot{u}_c$, $\dot{w}_c$, and $\dot{\omega}_c$ are the derivatives of the commands. The signals are u_{bs}, w_{bs} and ω_{bs} are backstepping terms defined to cancel sign-indefinite terms in the stability analysis.

The inner loop control signals are

$$\begin{aligned} u_1 &= M(F_{nlin} - K_v\tilde{v}_b + \dot{v}_{bc} - v_{bbs}) \qquad (17) \\ u_2 &= J(M_{nlin} - K_\omega\tilde{\omega} + \dot{\omega}_c - \omega_{bs}) \qquad (18) \end{aligned}$$

with the thrust vector defined as

$$T = \begin{bmatrix} L_f \\ L_m \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

where $\tilde{v}_b = v_b - v_{bc}$, $\tilde{\omega} = \omega - \omega_c$, and K_v and K_ω are positive definite matrices.

With this choice of the control signal and the fact that $\dot{\tilde{v}}_b = \dot{v}_b - \dot{v}_{bc}$ and $\dot{\tilde{\omega}} = \dot{\omega} - \dot{\omega}_c$, the dynamics of the tracking errors are

$$\begin{aligned} \dot{\tilde{u}} &= -K_u\tilde{u} - u_{bs} \\ \dot{\tilde{w}} &= -K_w\tilde{w} - w_{bs} \\ \dot{\tilde{\omega}} &= -K_\omega\tilde{\omega} - \omega_{bs} \end{aligned}$$

VIII. SIMULATION RESULTS

Figures 4–8 present the results from about 300 seconds of a simulated mission, during which the vehicle navigates around a box defined by the following four (N,E) corners: (0,-20), (10,-20), (10,-10), and (0,-10). The vehicle traverses the trajectory connecting the box corners. The 2D position plot is shown in Figure 4. The great trajectory repeatability and tracking of the desired path is observable. The transition at the corners can be improved by adding another controller which will command zero speed when the vehicle reaches each corner. In that situation, the vehicle would yaw to the appropriate angle and then resume its path. The position plot

showing north, east, down, and altitude position versus time is shown in Figure 5, while the attitude (roll, pitch, yaw) plot is shown in Figure 6. Vehicle's horizontal and vertical velocities are shown Figure 7, while the angular rates (roll, pitch, yaw rate) plot is shown in Figure 8. The vehicle is maneuvering in three dimensions, as the actual depth is distinct for each edge of the box (see plot three in Figure 5). The vehicle is programmed to maintain the altitude from the bottom as opposed to depth. Excellent tracking performance can be observed since the actual vehicle states (blue lines) converge to the desired ones (black and red lines). In addition to navigating to the four desired corners, the vehicle is conforming to the bowl shape of the TRANSDEC pool (a model of the pool is imported in the simulator). The vehicle is adjusting its pitch going up and down the pool and its adjusting its roll when going along the side of the bowl (see plot one and two in Figure 6). The desired distance from the bottom (altitude) is maintained within of ± 0.05 m of the desired altitude of 1.5 m as shown in the fourth plot in Figure 5.

IX. TRANSDEC Testing Results

Figures 9–13 present the results of a similar in length portion of an actual in-water mission, during which the vehicle navigates around a box in the TRANSDEC pool. This mission had the same requirements as the mission that was run by the simulator. The results also very closely resemble the simulated results which validates our approach. Again, the 2D position plot shown in Figure 9. The position plot showing north, east, and down position versus time is shown in Figure 10, while the attitude (roll, pitch, yaw) plot is shown in Figure 11. Vehicle's horizontal and vertical velocities are shown Figure 12, while the angular rates (roll, pitch, yaw rate) plot is shown in Figure 13. The vehicle is maneuvering in three dimensions, as the actual depth is distinct for each edge of the box. Great trajectory tracking performance can be noticed since the vehicle well maintained its track-line. It is worth to mention that this is greatly desired capability, an example being ship-hull inspection mission, since 100 percent ship-hull coverage is required. Excellent tracking performance can be observed, for instance, maximum altitude variations were around ± 0.2 m; the vehicle, as it was supposed changed its roll and pitch in order to accomplish the mission goal of translating while maintaining 1.5 m distance from the TRANSDEC bowl.

X. Conclusion

This paper has discussed the design and derivation of a command filtered, vector backstepping approach to design a stable translational and attitude controller (i.e., $y = [x(t), y(t), d(t), \psi(t), \theta(t)]^\top$) applicable to an underactuated UUV. The mission scenario specifies the position and attitude commands which are command filtered to produce inputs (together with their derivatives) for the outer loop and middle loop controllers. The commands, such as horizontal and vertical velocities (u_c,w_c) and angular rates (p_c,q_c,r_c) are generated by the outer and middle loop, command filtered, and are inputs (together with their derivatives) to the velocity and angular

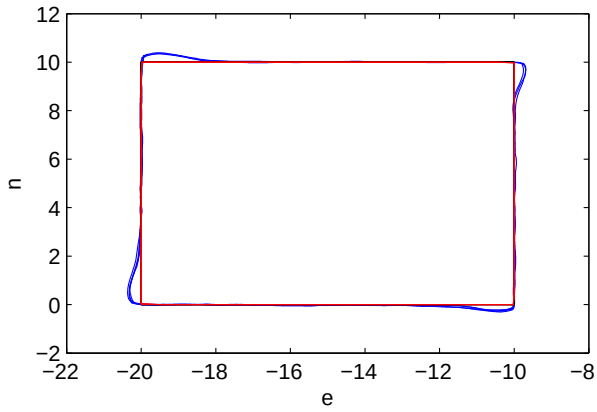

Fig. 4. 2 D Position vs. Time: Blue line is the actual vehicle trajectory, black line is the command, and the red line is filtered command

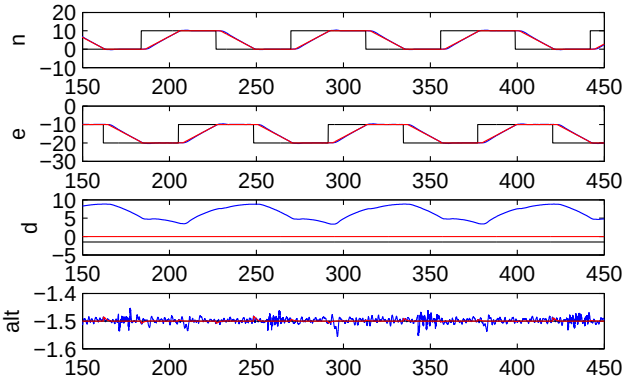

Fig. 5. North, East, Down, and Altitude Position vs. Time: Blue line is the actual vehicle trajectory, black line is the command, and the red line is filtered command

rate inner loop controllers. The inner loop determines the appropriates thrust forces. The article has presented both the control law derivation, the simulation results, and the actual in-water results. The actual in-water results very closely resemble the simulated mission for all control modes tested. This paper stresses the comparison between the simulated and actual in-water mission performed by the UUV and advocates the use of great simulation tool that was developed.

The plan for future is to design additional outer loop controllers to increase the vehicle maneuverability and capability. The outer loop controllers are defined for different vectors of outer loop control variables as specified by the vector of outputs y.

Acknowledgment

The authors gratefully acknowledge the Office of Naval Research (ONR) and Space and Naval Warfare Systems Center San Diego's (SSC-SD) In-house Independent Research (ILIR) Program for funding V. Djapic'c PhD work and the work described in this paper.

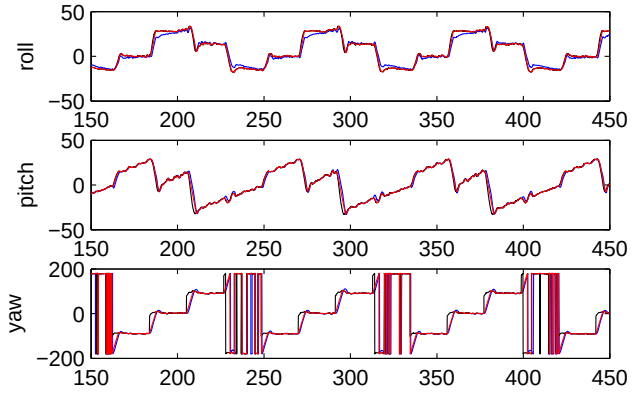

Fig. 6. Attitude vs. Time: Blue line is the actual vehicle attitude, black line is the command, and the red line is filtered command

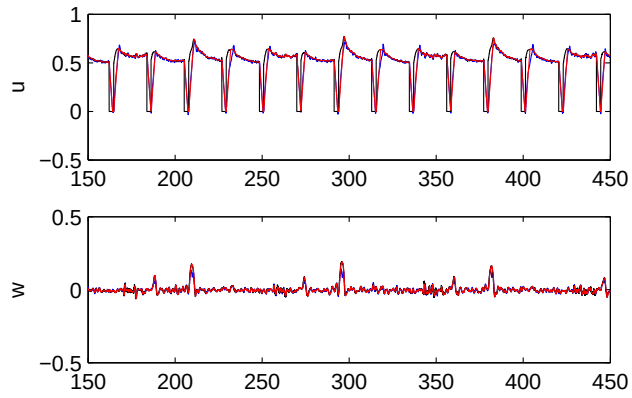

Fig. 7. Velocities vs. Time: Blue line is the actual vehicle velocity, black line is the command, and the red line is filtered command

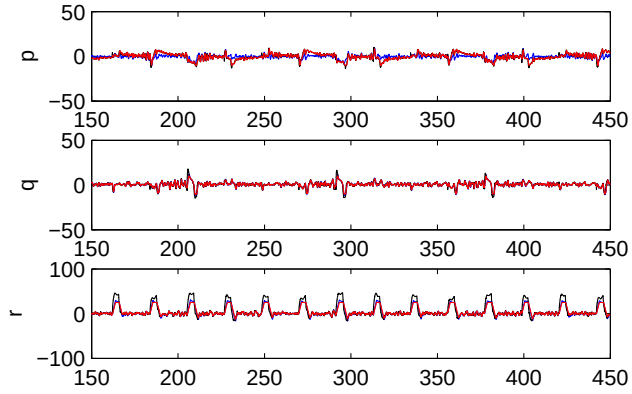

Fig. 8. Angular Rates vs. Time: Blue line is the actual vehicle angular rate, black line is the command, and the red line is filtered command

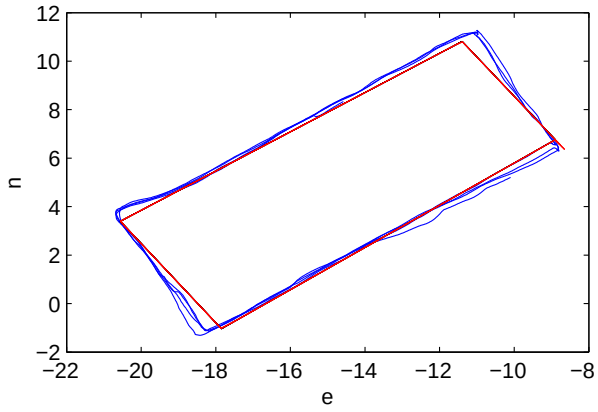

Fig. 9. 2 D Position vs. Time: Blue line is the actual vehicle trajectory, black line is the command, and the red line is filtered command

REFERENCES

[1] V. Djapic, J. A. Farrell "3D Command Filtered Backstepping" will be soon submitted to IEEE Trans. on Automatic Control or similar journal 2007.

[2] M. Breivik, T. I. Fossen, "Principles of Guidance-Based Path Following in 2D and 3D," Conference on Decision and Control, December 2005.

[3] J. A. Farrell, M. Polycarpou, M. Sharma, W. Dong, "Command Filtered Backstepping," Submitted to IEEE Trans. on Automatic Control, March 2007.

[4] J. A. Farrell, M. M. Polycarpou. "Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches," Hoboken, NJ: John Wiley, 436 pp., 8 chapters, 2006.

[5] J. A. Farrell, M. Polycarpou, M. Sharma, "Adaptive backstepping with magnitude, rate, and bandwidth constraints: Aircraft longitude control," American Control Conference, Denver, CO, June 2003, 3898-3903, 2003.

[6] J. A. Farrell, M. Polycarpou, M. Sharma, "On-Line Approximation Based Control of Uncertain Nonlinear Systems with Magnitude, Rate and Bandwidth Constraints on the States and Actuators," American Control Conference, Boston, MA, June 2004, 2557-2562, 2004.

[7] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, Nonlinear and Adaptive Control Design, John Wiley Sons, Inc, New York, 1995.

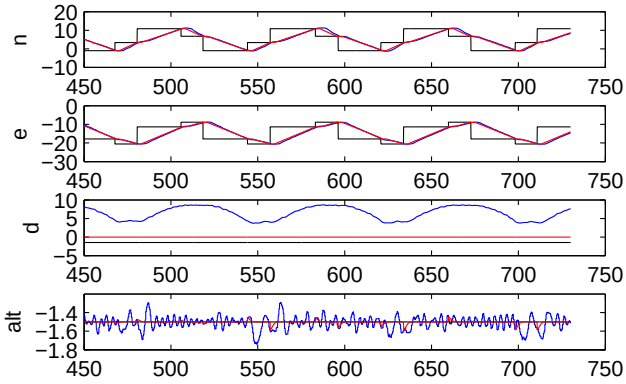

Fig. 10. North, East, Down, and Altitude Position vs. Time: Blue line is the actual vehicle trajectory, black line is the command, and the red line is filtered command

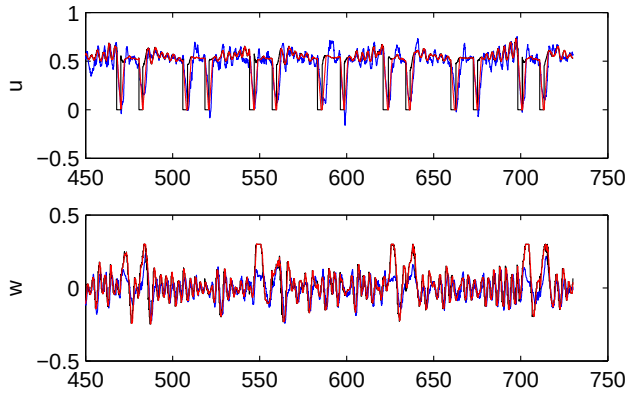

Fig. 12. Velocities vs. Time: Blue line is the actual vehicle velocity, black line is the command, and the red line is filtered command

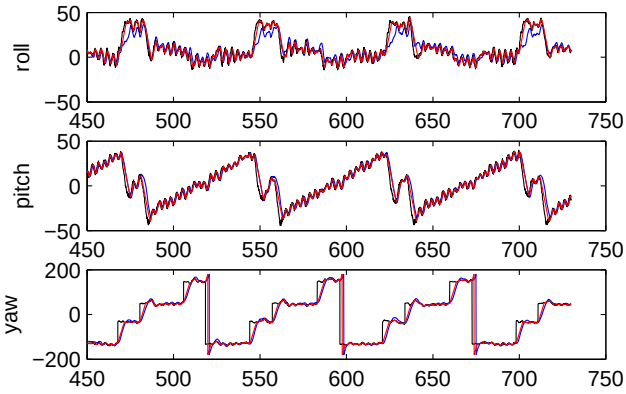

Fig. 11. Attitude vs. Time: Blue line is the actual vehicle attitude, black line is the command, and the red line is filtered command

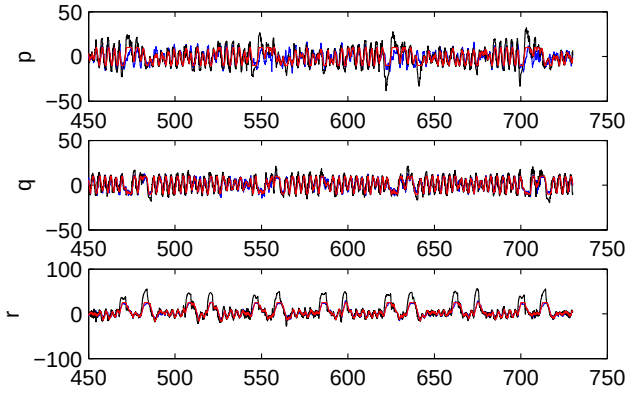

Fig. 13. Angular Rates vs. Time: Blue line is the actual vehicle angular rate, black line is the command, and the red line is filtered command